Honeywell

Flight Scheduling Optimization

Project Title:
Predictive Scheduling & Impact Analysis for Busy Airports (Mumbai - BOM)

Author:
Rishi Guptha Neelisetty

Candidate ID:
16845317

Mail ID:
rishiguptha45@gmail.com

Submission Overview

This document presents a complete project proposal, technical approach, prototypes, and evaluation plan for analyzing and improving flight schedules at busy airports (example: Mumbai - BOM) using open-source AI tools and flight-tracking data (Flightradar24 / FlightAware). The deliverable is a working pipeline that ingests one week of flight data, performs exploratory and causal analysis, builds models for delay prediction and cascading-impact estimation, and provides an NLP-driven query interface for stakeholders.

1. Proposed Solution

1.1 Problem Summary

Busy airports face runway capacity and scheduling limits. Peaks around similar hours create queuing, and small disruptions cascade into widespread delays. Stakeholders need data-driven scheduling guidance: best times to schedule arrivals/departures, slots to avoid, and which flights cause the biggest downstream disruption.

1.2 High-Level Solution

We propose a data-driven system that ingests flight schedule and movement data, enriches it with weather and runway capacity, and applies machine learning plus graph-based simulations. The system predicts delays, identifies peak congestion

slots, and simulates cascading effects of disruptions. An NLP interface allows planners to query insights in plain language and run "what-if" scheduling scenarios.

1.3 How It Addresses the Problem

- Highlights high-risk time slots so planners can avoid congestion.
- Enables what-if experiments (e.g., shifting a flight by X minutes) to see delay impacts.
- Ranks flights by cascading influence, guiding where to add buffers or give priority.

1.4 Innovation & Uniqueness

- Goes beyond delay prediction by combining flight-tracking data with cascade simulations.
- Provides an interactive, NLP-driven scheduler for intuitive decision-making.
- Introduces influence scores to identify flights most critical to network-wide delays.

2. Technical Approach

2.1 Data Sources

- Flightradar24 / FlightAware: Scheduled vs actual times, aircraft type, airline, origin/destination, flight status.
- Weather API: Hourly METAR data (visibility, wind, rain).
- Airport Notes (if available): Runway configuration, NOTAMs, operational restrictions.

2.2 Technologies & Tools

- Languages: Python (main), JavaScript (for frontend).
- Libraries: pandas, NumPy, scikit-learn, LightGBM,XGBoost, networkx (graphs), Hugging Face (NLP), React (UI), Matplotlib (plots).
- Storage: Parquet / SQLite (prototype), Postgres (production).
- Deployment: FastAPI backend + React UI

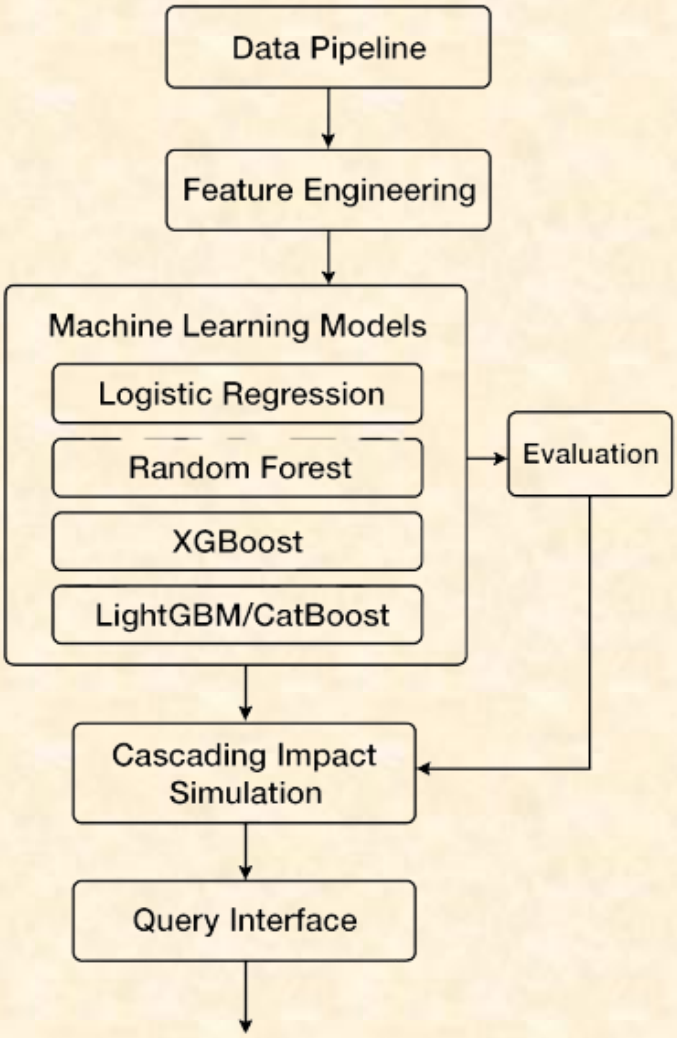
2.3 Data Pipeline (High-Level Flow)

1. Ingestion: Collect one week of Flightradar24 data.
2. Preprocessing: Clean duplicates, normalize times, compute turnaround/taxi times, calculate delays, create 15–30 min slots.
3. Enrichment: Merge with weather, runway, and airline details.
4. Exploratory Analysis: Heatmaps of busy slots, delay distributions by airline and time.
5. Delay Model: Train ML model to predict expected delay with confidence ranges.

6. Cascading Impact Model: Build graph of flight connections, simulate how delays spread, calculate influence score per flight.
7. Schedule Optimizer: Suggest time shifts to minimize total delays while respecting runway limits.
8. NLP Query Interface: Allow natural language queries like “Which is the busiest time to land?”

2.4 Models and Methods

- Delay Predictor: Uses time of day, weekday, airline, aircraft type, previous delay, congestion, and weather. (LightGBM with quantile regression).
- Propagation Simulation: Flights as nodes, edges = aircraft/crew/passenger connections. Simulates runway queue and delay spread.
- Influence Score: Measures how much extra delay a single flight causes downstream.
- Schedule Tuner: Adjusts flight times (via heuristic optimization) and checks impact on overall delay.



2.5 Interface & UX

- Dashboard: Charts for delay trends, slot congestion, and major causes.
- What-if Panel: User can shift a flight's time and instantly see system-wide delay changes.
- NLP Box: Type a question in plain English (e.g., "Show busiest slots on today evening") → system generates results and visuals.

3. Feasibility & Viability

3.1 Feasibility Analysis

- Data availability: Flightradar24 and FlightAware provide schedule and status information (one-week sample is feasible to extract). Weather and runway availability are accessible publicly.
- Technical feasibility: The stack listed is standard and implementable within a typical data-science environment.

3.2 Challenges & Risks

- Data completeness: Missing tail numbers or incomplete connection data makes the construction of dependency graph noisy.
- Accuracy of simulation: Modeling runway operations precisely requires detailed service-time distributions and accurate runway configuration history.
- API / scraping legal constraints: Respect terms of use of data providers.
- Computational cost: Full combinatorial schedule tuning can be expensive for many flights.

3.3 Mitigation Strategies

- Use probabilistic edges where explicit dependencies are missing, estimated from historical co-occurrence patterns.
- Start with coarse time granularity (15/30 mins) for fast prototyping; refine to minute-level for final runs on smaller subsets.
- Use sampling and heuristic optimization (local search, greedy) instead of exhaustive search.
- Document data provenance and respect API rate limits.

4. Research & References

- Flightradar24 — https://www.flightradar24.com/ (airport and schedule pages)
- FlightAware — https://www.flightaware.com/
- Research literature: Delay propagation modeling, network-centric airline delay studies (e.g. papers on delay propagation, queuing models for runway operations)
- Tools & Libraries: LightGBM, NetworkX, scikit-learn, Streamlit

- Kaggle dataset used for reference : https://www.kaggle.com/datasets/usdot/flight-delays/data

5. Prototype Website for Proposed Model

(The data displayed on this website is assumed/sample data. It is not calculated in the prototype stage. Once the model is fully developed, the results will be integrated into the website.)

Website URL: flightOPs
https://flight-delay-predict-cspk.bolt.host/

GitHub Repository (currently under development): Repo
https://github.com/rishiguptha54/FlightOpsPro_45

Screenshots of Codes Snippets/Results & Website

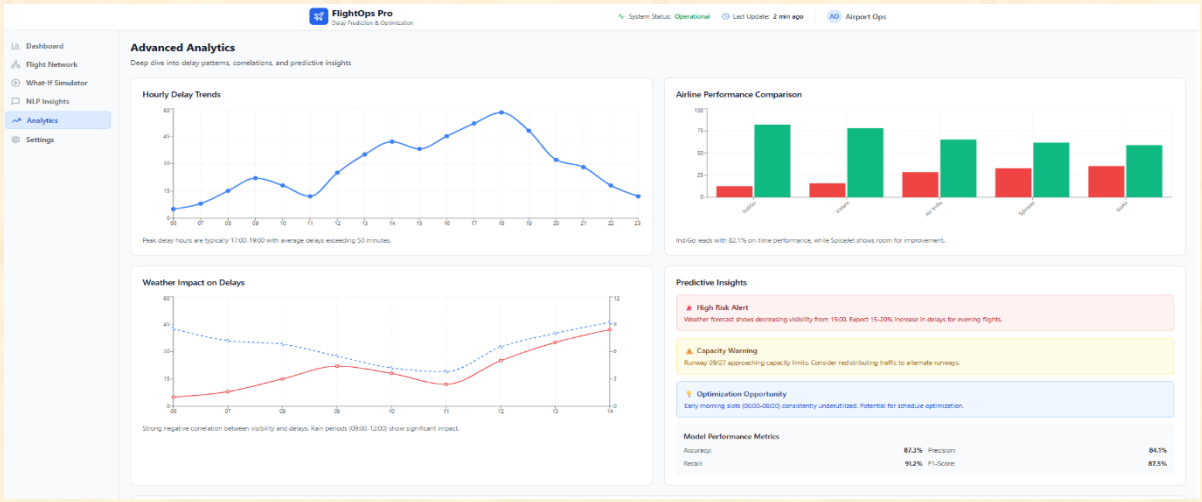
```
[55]
    # Define the base models
    estimators = [
        ('lr', log_reg),
        ('rf', rand),
        ('xgb', xgb),
        ('lgbm', lgbm),
        ('catboost', catboost)
    ]

    # Define the stacking classifier
    stk = StackingClassifier(estimators=estimators, final_estimator=LogisticRegression())

    # Train the stacking classifier
    stk.fit(X_train, y_train)

    # Make predictions
    y_pred_stk = stk.predict(X_test)

    # Print classification report
    print(classification_report(y_test, y_pred_stk))
```

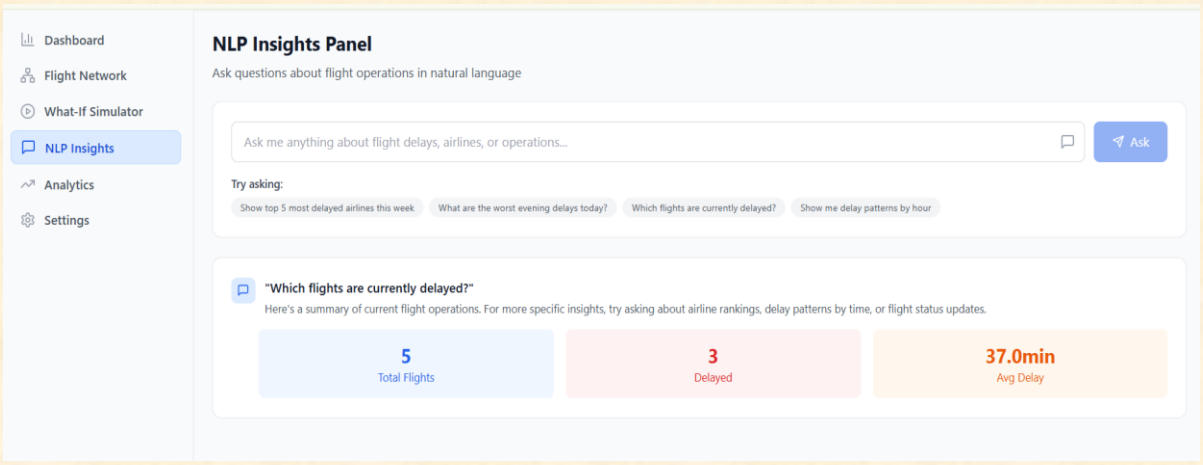
Here is a summary of the performance metrics for each model based on the classification reports:

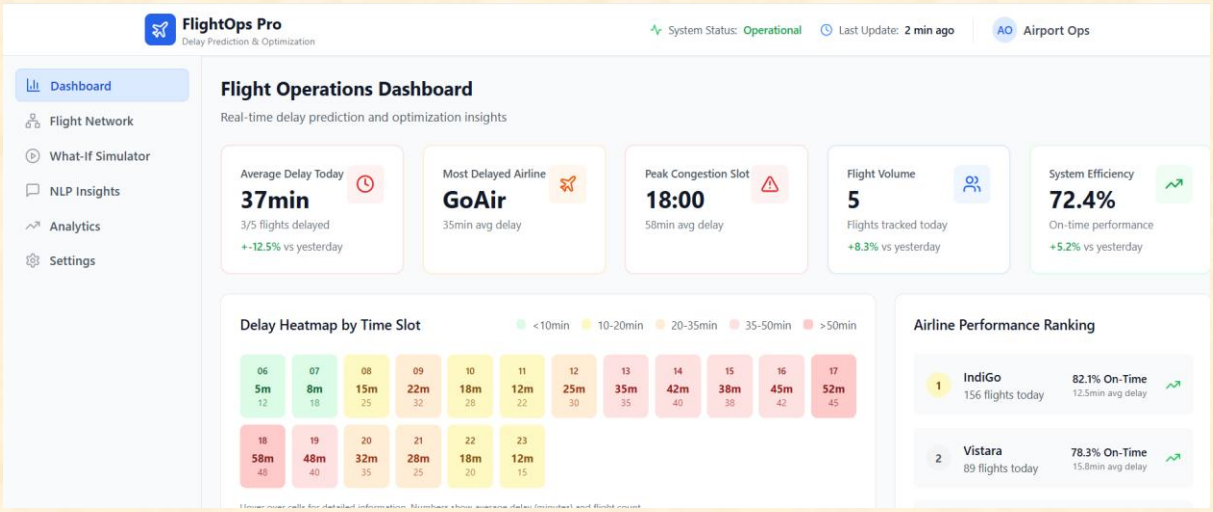
Model	Precision (Class 0)	Recall (Class 0)	F1-score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-score (Class 1)	Accuracy
Logistic Regression	1.00	1.00	1.00	0.98	0.94	0.96	1.00
Random Forest	1.00	1.00	1.00	1.00	1.00	1.00	1.00
XGBoost	1.00	1.00	1.00	0.00	0.00	0.00	1.00
LightGBM	1.00	1.00	1.00	0.00	0.00	0.00	1.00
CatBoost	1.00	1.00	1.00	0.91	0.82	0.86	1.00

Based on these metrics, the Random Forest classifier appears to have performed the best, achieving perfect precision, recall, and f1-score for both classes on the test set. CatBoost also performed well, especially considering the imbalanced nature of the data (very few delayed flights). Logistic Regression had a slightly lower recall and f1-score for the delayed class compared to Random Forest. XGBoost and LightGBM struggled to identify the delayed flights, resulting in a precision and recall of 0.00 for class 1.

Keep in mind that due to the extreme class imbalance, models that predict '0' for all instances will still show a high overall accuracy. It's important to focus on the precision, recall, and f1-score for the minority class (class 1 - delayed flights) when evaluating these models.


FlightOps Pro
Delay Prediction & Optimization
System Status: Operational
Last Update: 2 min ago
AO Airport Ops
Dashboard
Flight Network
What-If Simulator
NLP Insights
Analytics
Settings
Advanced Analytics
Deep dive into delay patterns, correlations, and predictive insights
Hourly Delay Trends
Peak delay hours are typically 17:00-19:00 with average delays exceeding 50 minutes.
Airline Performance Comparison
IndiGo leads with 82.1% on-time performance, while SpiceJet shows room for improvement.
Weather Impact on Delays
Strong negative correlation between visibility and delays. Rain periods (09:00-12:00) show significant impact.
Predictive Insights
High Risk Alert
Weather forecast shows decreasing visibility from 15:00. Expect 15-20% increase in delays for evening flights.
Capacity Warning
Runway 09/27 approaching capacity limits. Consider redistributing traffic to alternate runways.
Optimization Opportunity
Early morning slots (06:00-08:00) consistently underutilized. Potential for schedule optimization.
Model Performance Metrics
Accuracy: 87.3%
Precision: 84.1%
Recall: 91.2%
F1-Score: 87.5%


Dashboard
Flight Network
What-If Simulator
NLP Insights
Analytics
Settings
NLP Insights Panel
Ask questions about flight operations in natural language
Ask me anything about flight delays, airlines, or operations...
Ask
Try asking:
Show top 5 most delayed airlines this week
What are the worst evening delays today?
Which flights are currently delayed?
Show me delay patterns by hour
"Which flights are currently delayed?"
Here's a summary of current flight operations. For more specific insights, try asking about airline rankings, delay patterns by time, or flight status updates.
5
Total Flights
3
Delayed
37.0min
Avg Delay


FlightOps Pro
Delay Prediction & Optimization
System Status: Operational
Last Update: 2 min ago
AO Airport Ops
Dashboard
Flight Network
What-If Simulator
NLP Insights
Analytics
Settings
Flight Operations Dashboard
Real-time delay prediction and optimization insights
Average Delay Today
37min
3/5 flights delayed
+-12.5% vs yesterday
Most Delayed Airline
GoAir
35min avg delay
Peak Congestion Slot
18:00
58min avg delay
Flight Volume
5
Flights tracked today
+8.3% vs yesterday
System Efficiency
72.4%
On-time performance
+5.2% vs yesterday
Delay Heatmap by Time Slot
<10min
10-20min
20-35min
35-50min
>50min
06 5m 12
07 8m 18
08 15m 25
09 22m 32
10 18m 28
11 12m 22
12 25m 30
13 35m 35
14 42m 40
15 38m 38
16 45m 42
17 52m 45
18 58m 48
19 48m 40
20 32m 35
21 28m 25
22 18m 20
23 12m 15
Airline Performance Ranking
1 IndiGo
156 flights today
82.1% On-Time
12.5min avg delay
2 Vistara
89 flights today
78.3% On-Time
15.8min avg delay

Thank You